Peripheral Servicing

The present invention relates to computer systems, and more particularly to scheduling of servicing of peripherals calling for attention in such systems.

In many computer systems, there is processing unit of some kind (which we will call a central processing unit, CPU) and a plurality of peripherals (which term may include any device outside the CPU core) which compete to be serviced by the CPU. A typical example is in a message switching system where there is a plurality of channels with messages arriving on some channels and being sent out on others. For each channel, there will be a buffer. Incoming data accumulates in the buffers of receiving channels, and has to be transferred from those buffers to a central memory; outgoing data is fed out to the outgoing channels from the buffers of those channels, and the outgoing data has to be passed to those buffers from the central memory.

The data flows between the buffers and the central memory are controlled by the CPU. These data flows (more specifically, the data flow rates for the various channels) are likely to be highly variable. Poor scheduling control by the CPU may result in the buffers of outgoing channels underflowing and, more seriously, in the buffers of incoming channels overflowing, which can lead to data loss.

Similar scheduling problems can arise more widely, in other types of computer system.

One technique for scheduling in such situations is to assign priorities to the various devices. The CPU effectively scans or polls the devices, starting with the highest priority device. As soon as the CPU finds a device requiring servicing, it services the device; it then resumes the polling of the devices from the beginning (ie from the highest priority device) again.

If the priority sequence is fixed, it is usually hard-wired (ie defined by hardware). If it is desired to make the priority sequence adjustable, then it can be implemented by software, eg using a table which defines the priority sequence of the various devices. However, that involves a certain amount of CPU overhead and delay, and is generally not preferred.

A drawback of such priority based techniques is that a low priority device may be starved of service indefinitely, if higher priority devices keep demanding servicing before the CPU's scanning can reach the low priority device.

An alternative technique has therefore been developed, in which the CPU polls cyclically through the devices. Each time the CPU encounters a device needing servicing, it services that device; it then resumes its polling of the devices from that device onwards. This technique is termed the round robin technique.

In the round robin technique, all devices effectively have equal priority. There is therefore no additional flexibility to be gained by implementing it by software. Since software implementation involves additional CPU overheads and delays, the round robin technique is therefore normally implemented by hardware, using a pair of registers and associated logic circuitry. There is a device status register, with one bit for each device (ie incoming or outgoing channel). When a device needs servicing, it sets this bit (and clears it when it no longer needs servicing). There is also a round robin register, with the same number of bits as the device status register.

A single bit is set in the round robin register, and determines which device the CPU services. Each time the CPU finishes servicing a device, the bit in the round robin register is advanced to the position of the next set bit in the device status register (both registers are effectively cyclic). If no devices call for servicing, the device status register will have no bits set; the round robin register is then normally maintained with its single bit in the position matching the last device to be serviced; as soon as another device requires servicing, the corresponding bit is

set in the device status register, and the bit in the round robin register is immediately advanced to the corresponding position.

The drawback of the round robin technique is precisely that all devices have effectively equal priorities. If a particular device has a high activity, then when the CPU has finished servicing it, it is desirable for it to be serviced again fairly soon. But with the round robin technique, all the other devices are effectively polled before the first device is reached again. If many of those other devices happen to require servicing, then they will all be serviced before the first device is reached again, even though their servicing could in fact be safely postponed, and the first device may by that time have overflowed.

EP-A-0339782 describes a system for achieving fixed priority, round robin or some combination of the two by means of a counter with programmable limits. The counter is incremented or decremented synchronously with the available time slots. US-A-5072363 is a variation on the latter technique in that it achieves different levels of service by using a counter to change the arbitration behaviour in a deterministic manner.

The general object of the present invention is to improve scheduling to alleviate or overcome the problems in the prior art. This is achieved by providing a computer system in which scheduling provides advantages of both the priority and round robin techniques and the devices can request servicing at different priority levels depending on their dynamic state, this state being internal to the device and not related to any central bus in the system. .

The invention is defined by the following claims.

A feature of the invention is that each peripheral device has associated with it (which may include a component integrated with the device) means arranged to produce a priority level signal which is indicative of the urgency with which that device needs servicing. For example, if three priority levels are used in a message

switching system, a receiving device may produce a priority level 3 signal (low priority) when its buffer is not empty, a priority level 2 signal (medium priority) when its buffer is half full, and a priority level 1 signal (high priority) when its buffer is nearly full. A transmitting device may produce a priority level 3 signal (low priority) when its buffer is nearly full, a priority level 2 signal (medium priority) when its buffer is half full, and a priority level 1 signal (high priority) when its buffer is nearly empty. Of course, these are just examples; fewer (2 priority levels) or more may be employed, and the conditions for signalling the various levels may vary from device to device. In particular, some devices which receive data at a higher rate, or have a smaller buffer may switch to a higher priority level at lower levels of buffer occupancy (or emptiness in the case of an output device) than others. There will be no signal when the buffer is completely empty for a receiving device, or completely full for a transmitting device.

It is preferred that a device should only produce a single priority level signal at a time. However, it is possible that when a device produces a signal of the appropriate priority level, it may produce signals of lower priority level as well (so for example a level 2 priority signal will be automatically accompanied by a level 3 priority signal). This will not have a significant effect on the way in which the present system operates.

It will of course be realized that not all devices may need to produce priority level signals of all levels. For instance, a transmitting device may not be able to produce a priority signal of as high a level as a receiving device, because the consequences of underflow in a transmitting device buffer will often be less serious than the consequences of overflow in a receiving device buffer.

The devices pass their priority level signals to a device status register means which stores these signals. Associated with this means there is a round robin register means. In a standard round robin register, there is a marker bit which is advanced, each time a device is serviced, from that device to the next device needing servicing. In the present round robin register means, the marker bit is

advanced from the current device to the next device at the highest active priority level.

The device status register means preferably consists of a separate device status register for each of the possible priority levels, and the round robin register means preferably consists of a corresponding or associated round robin register for each device status register, each containing the standard single bit.

When a device produces a priority signal, it sets the corresponding bit in the corresponding device status register. The device status registers are monitored to identify the highest priority level register with at least one bit set. The associated round robin register for that level is then used to determine which device the CPU should service next, and its bit is advanced to the next device (if any) at that priority level, ie to the next device in the corresponding device status register.

Thus in the present system, devices are normally processed in round robin fashion. However, the system also operates at a plurality of priority levels. Only devices at the highest active priority level are eligible for servicing.

However, no device is ever locked out indefinitely. Lock-out of devices is prevented in two ways. As the servicing of a device becomes more urgent, the device priority level is raised; and once all devices at the highest active priority level are fully serviced, the priority level drops to the next level down, and servicing of devices at that level recommences. It should be noted that in this latter situation, the servicing of devices at the new lower level recommences at the point where it was left off when the next priority level up became active. In other words, the position of the bit in each round robin register is frozen when the system enters a higher priority level. In fact, the devices will only enter the higher priority levels when the system is heavily loaded. As described in our concurrently filed application bearing reference PDC/20020, incorporated herein by reference, the system may include means for allocating more system resources to the processor in response to detection that one or more devices are becoming in need

of urgent servicing.

A computer system embodying the invention will now be described, by way of example, with reference to the drawings, in which:

Fig. 1 is a simplified block diagram of the system; and

Fig. 2 is a simplified block diagram of the scheduling means.

Referring to Fig. 1, the system comprises a set of devices 10-1 to 10-n, a CPU 11, and a memory 12. The devices 10 are communication devices, devices 10-1 and 10-2 being transmitting devices receiving data from the CPU and device 10-n being a receiving device feeding data to the CPU. It will be understood that the arrowheads on the couplings between the devices and the CPU show the direction of data flow; there are control signals flowing from all devices to the CPU and vice versa.

Each device includes a buffer BUFF, and priority setting means which monitors the occupancy of the buffer and generates a priority level signal accordingly. For a transmitting device such as devices 10-1 and 10-2 the priority setting means produces a priority level 3 signal (low priority) when its buffer is nearly full, a priority level 2 signal (medium priority) when its buffer is half full, and a priority level 1 signal (high priority) when its buffer is nearly empty. For receiving devices such as device 10-n the priority setting means produces a priority level 3 signal (low priority) when its buffer is not empty, a priority level 2 signal (medium priority) when its buffer is half full, and a priority level 1 signal (high priority) when its buffer is nearly full. There will be no signal when the buffer is completely empty for a receiving device, or completely full for a transmitting device.

The CPU 11 includes scheduling means 13 which receive the priority signals from the devices 10 and determine which device the CPU will service next.

The scheduling means are shown in more detail in Fig. 2.

There are 3 device status registers 20-1, 20-2, and 20-3, one for each of the 3 priority levels. Each device status register has the same number of positions (bits) as the number of devices 10, one position for each device. The device status registers are fed with the priority level signals from the devices, so that when a device produces a priority level signal, the corresponding position in the corresponding device status register is set (and cleared when the device clears its priority level signal). The 3 device status registers thus together store the priority levels of the devices.

There is also a round robin register means consisting of a set of round robin register 21-1 to 21-3, one for each of the device status registers. Each of the round robin registers contains a single bit, whose position corresponds to a set bit in the associated device status register. The position of this bit in a selected one of the round robin registers determines which device will next be serviced by the CPU. When a device is so serviced and its priority level signal is cleared, the bit in that round robin register is advanced to correspond to the next set bit in the corresponding device status register.

The selection of the round robin register in the round robin register means for this is performed as follows.

A priority determining circuit 22 is fed from all the device status registers 20, and determines what the highest active priority is. If there is any bit set in the highest priority device status register 20-1, the highest priority is 1; if there is no bit set in the highest priority device status register 20-1, but any bit is set in the middle priority device status register 20-2, the highest priority is 2; if there is no bit set in the highest priority device status register 20-1 or the middle priority device status register 20-2 but any bit is set in the low priority device status register 20-3, the highest priority is 3. (If none of the device status registers has any bit set, then there is no device requiring service from the CPU.)

The priority determining circuit 22 enables the corresponding round robin register 21. This allows the set bit in the enabled register to advance to the position corresponding to the next set bit in the corresponding device status register, as described above.

The priority determining circuit 22 also controls a gate circuit 23, which couples the appropriate one of the round robin circuits 21 to an overall round robin circuit 24. This overall round robin circuit therefore contains a single bit, whose position matches that of the bit in the round robin circuit with the highest active priority. This information is conveyed to the CPU 11 at the appropriate time, for example in response to a read of this register by the CPU, or following an interrupt scheduled to ensure regular servicing of peripherals, and the CPU thereupon services the device corresponding to this set bit.